

جلسه هشتم: دیپلوی سیستم UTAXI روی سرور ابرآروان

جلسه هشتم پروژه UTAXI به دیپلوی (Deploy) سیستم روی سرور ابرآروان اختصاص یافت. پس از توسعه، تست و بهینه‌سازی سیستم در جلسات گذشته، تیم تصمیم گرفت که سرویس را به یک محیط ابری منتقل کند تا کاربران بتوانند از آن در محیط واقعی استفاده کنند. ابرآروان به‌عنوان پلتفرم انتخاب‌شده برای میزبانی، به دلیل سرعت بالا، امنیت مناسب و هزینه مقرون‌به‌صرفه انتخاب شد.

جلسه با مدیریت امین یوسفی آغاز شد و وی توضیح داد که هدف این جلسه آماده‌سازی زیرساخت سرور، نصب سرویس‌های موردنیاز و انجام مراحل دیپلوی سیستم UTAXI است. همچنین برنامه‌ریزی‌هایی برای مدیریت و نگهداری سرور پس از دیپلوی انجام شد.

بخش اول: انتخاب سرور و آماده‌سازی زیرساخت

در ابتدا، تیم در مورد انتخاب نوع سرور و مشخصات سخت‌افزاری مورد نیاز بحث کرد. ابرآروان به‌عنوان ارائه‌دهنده خدمات ابری انتخاب شد و یک سرور **Cloud Compute** در این پلتفرم راه‌اندازی شد. مشخصات سرور انتخاب‌شده عبارت بودند از:

CPU ✓ : دو هسته پردازشی

RAM ✓ : چهار گیگابایت حافظه رم

Storage ✓ : پنجاه گیگابایت فضای ذخیره‌سازی SSD

OS ✓ : Ubuntu 22.04

✓ پشتیبانی از **Docker** و قابلیت اجرای کانتینرها

پس از خرید و راه‌اندازی سرور، دسترسی **SSH** به اعضای تیم داده شد تا بتوانند مراحل نصب و پیکربندی را انجام دهند.

بخش دوم: نصب پیش‌نیازها و تنظیم سرور

برای اجرای سیستم UTAXI روی سرور، ابتدا پیش‌نیازهای نرم‌افزاری زیر نصب شدند:

```
sudo apt update && sudo apt upgrade -y
```

```
sudo apt install -y docker.io docker-compose nginx certbot python3-certbot-nginx sqlite3
```

Docker و Docker Compose: برای اجرای سرویس‌ها در کانتینرهای مجزا. ✓

Nginx: به‌عنوان **ریورس پراکسی** برای مدیریت درخواست‌های ورودی و تأمین امنیت سرور. ✓

Certbot: برای ایجاد و مدیریت **گواهی SSL** و تأمین ارتباطات امن. ✓

SQLite3: برای مدیریت پایگاه داده محلی در سرور. ✓

پس از نصب پیش‌نیازها، **فایروال سرور (UFW) تنظیم شد** تا فقط **پورت‌های ضروری** باز باشند:

```
sudo ufw allow OpenSSH
```

```
sudo ufw allow 80
```

```
sudo ufw allow 443
```

```
sudo ufw enable
```

بخش سوم: کانتینری سازی و اجرای سرویس ها

برای اجرای UTAXI روی سرور، **Docker Compose** مورد استفاده قرار گرفت. تیم یک فایل **docker-compose.yml** ایجاد کرد که شامل سرویس های مورد نیاز بود:

```
version: '3.8'
```

```
services:
```

```
  app:
```

```
    image: utaxi:latest
```

```
    restart: always
```

```
    ports:
```

```
      - "5000:5000"
```

```
    environment:
```

```
      - DATABASE_URL=sqlite:///utaxi.db
```

```
    volumes:
```

```
      - ./data:/app/data
```

```
  nginx:
```

```
    image: nginx:latest
```

```
    restart: always
```

```
    ports:
```

```
      - "80:80"
```

```
      - "443:443"
```

```
    volumes:
```

```
      - ./nginx.conf:/etc/nginx/nginx.conf
```

✓ سرویس **app** شامل برنامه اصلی UTAXI است که روی پورت ۵۰۰۰ اجرا می‌شود.

✓ سرویس **nginx** برای مدیریت درخواست‌ها و بهینه‌سازی امنیت سرور به کار گرفته شد.

✓ یک پایگاه داده **SQLite** برای ذخیره داده‌ها در محیط سرور تنظیم شد.

برای اجرای سرویس، تیم از دستور زیر استفاده کرد:

```
docker-compose up -d
```

بخش چهارم: تنظیم Nginx و صدور گواهی SSL

برای فراهم‌سازی ارتباط امن HTTPS، تنظیمات Nginx انجام شد و گواهی SSL رایگان از Let's Encrypt دریافت گردید:

```
sudo certbot --nginx -d myfreecourse.ir
```

✓ Nginx به‌عنوان لایه مدیریت درخواست‌ها تنظیم شد.

✓ SSL فعال شد تا ارتباطات ایمن شوند و کاربران بتوانند با اطمینان از سرویس استفاده کنند.

✓ دامنه **myfreecourse.ir** به سرور متصل شد تا کاربران بتوانند از طریق آدرس

<https://myfreecourse.ir/> به سرویس دسترسی داشته باشند.

بخش پنجم: تست و بررسی عملکرد سرور

پس از انجام مراحل دیپلوی، تیم چندین تست برای اطمینان از عملکرد صحیح سیستم انجام داد:

✓ **تست API:** بررسی عملکرد API ها و پاسخ‌دهی مناسب سرور.

✓ **تست بارگذاری (Load Testing):** ارسال درخواست‌های متعدد به سرور و بررسی میزان پاسخ‌دهی.

✓ **تست امنیتی:** بررسی فایروال و لاگ‌های سرور برای اطمینان از عدم وجود دسترسی‌های غیرمجاز.

✓ **تست تجربه کاربری:** ارزیابی استفاده از سیستم از طریق مرورگر و تعامل کاربران واقعی.

نتایج نشان داد که سیستم به‌درستی روی سرور ابرآروان اجرا شده و کاربران می‌توانند به سرویس دسترسی داشته باشند.

نتیجه‌گیری

جلسه هشتم با موفقیت انجام شد و سیستم UTAXI با موفقیت روی سرور ابرآروان دیپلوی گردید. تیم توانست با استفاده از **SSL، Nginx، Docker و SQLite**، یک زیرساخت امن و کارآمد برای میزبانی سرویس ایجاد کند. در جلسات آینده، تمرکز بر روی نظارت بر عملکرد سرور، بهینه‌سازی امنیت و افزایش مقیاس‌پذیری سرویس خواهد بود.